Changes in the Specification

Rewrite the paragraph beginning on page 3, line 24, as follows:

" The JVMPI represents a considerable advance over previous Java profiling solutions as it offers a standard, extendible interface, agreed through consultation with Java tools vendors and other companies such as IBM. It defines a tool for acquiring comprehensive profiling data. Typically, the different profiling tools will have varying aims and require different information. For example, one may merely dump profile data to a trace file, while another tool may process the data selectively and interactively, and present a more sophisticated view via a graphical user interface. Tool vendors may write to the JVMPI, rather than having to code custom-built hooks directly into the JVM itself. It allows them to access consistent interfaces and events on JVM implemtnations which have a JVMPI consistent with Sun's definition. Commercially available profiling tools include Jprobe from KL Group Inc., Canada and True Coverage from Compuware NuMega, America."

Rewrite the paragraph beginning on page 7, line 1, as follows:

" The profiler agent invokes an EnableEvent () function via "JVMPI_Interface->EnableEvent ()" in order to instruct the JVM to communicate a given event to the notify Event function upon occurrence of that even (step 75). Such events are then received by the profiler agent from the JVMPI (step 80)."

Delete the paragraph beginning on page 7, line 23, and extending through page 8, line 2.

Rewrite the paragraph beginning on page 10, line 9, as follows:

" While the profiler agent communicates remotely with the JVM 110, it does so via a direct entry-point. In other words, the JVM 110 itself has been specially modified to support the Profiler Agent 140. The JVMPI and all the advantages it provides have been dispensed with in

this solution. Although the agent code does not have to be ported to the embedded device, it does have to be compiled and debugged by the tool writer in order to be compatible with the specific JVM implementation."

Rewrite the paragraph beginning on page 11, line 25, as follows:

" After doing this work, I concluded this scenario would not achieve its primary aim of transparency. The profiling agent would realize that the events it receives from JVM 226 had indeed come from its local machine 207 rather than the true originator, JVM 200. This is because JNI interface 226 is presented to the profiling agent, rather than a JVMPI interface as should be the case. I stopped the described work without revealing it publicly and proceeded with completion of the invention described herein."

Delete the entire paragraph beginning on page 12, line 5.

Rewrite the paragraph beginning on page 16, line 9, as follows:

" Figure 3 shows a remote profiling solution using a JVMPI according to work I did leading up to the present invention."

Rewrite the paragraph beginning on page 18, line 14, as follows:

" Specific event types are requested over the first connection 11a, 11b and returned to the main UJA thread. Events are then transmitted in the reverse direction to the UJA and interface functions are processed to enable this and to enable the remote profiler agent to gather additional information."

Rewrite the paragraph beginning on page 19, line 1, as follows:

" In the preferred embodiment, those events which do not need an immediate response from the profiler agent are buffered and sent asynchronously to the profiler agent buffer receiving

thread 356 (arrow 14b), thus reducing network traffic. Some events however will have to be sent synchronously in order that the JVM does not terminate prematurely as a result of a delayed response from the profiler agent. According to a preferred embodiment of the invention, JVMPI_EVENT_GC_FINISH, JVMPI_EVENT_GCSTART, JVMPIE_EVENT_JVM_SHUT_DOWN, JVMPI_EVENT_THREAD_END and JVMPI_EVENT_THREAD_START reuire synchronous transmission.

Rewrite the paragraph beginning on page 20, line 11, as follows:

" Note, the data transmitted between thread 365 and thread 356 is preferably compressed before transmission. It will be appreciated that the JVM may generate a large number of events and that both the buffering and data compression help prevent the network from being overwhelmed."

Rewrite the paragraph beginning on page 24, line 25, as follows:

" According to the preferred embodiment, the structure of control data packets are of big-endian format. They are also similar to those defined by Sun Microsystems Corporation in the Java Debug Wire Protocol (JDWP). Some key differences, however, are that the USA control packet flows are synchronous and do not have to be paired with unique id fields. The id field is provided for other purposes, described below. The layout of these packets is shown in Figure 6b."

Rewrite the paragraph beginning on page 29, line 8, as follows:

" Once again there is also a data field 840 of variable length. This contains the information requested by a corresponding control. For example, in response to a command, 'Query JVPI version' int jvmpiVersion will be returned. This is the JVMPI version number as returned from JVMPI inteface version function ('jint version').